

REMARKS

Claims 2-8, 10-16 and 18-24 are pending in the above-referenced application (this "Application").

Claims 2-8, 10-16 and 18-24 have been rejected in this Application.

No claims have been amended in this response.

Reconsideration of the claims is respectfully requested. The claims are listed in APPENDIX "A" in their current form, for the Examiner's easy reference. The Applicant believes that this Application is in condition for appeal.

35 U.S.C. § 103(a) -- Obviousness

In the Office Action, The Examiner rejected Claims 2-8, 10-16, and 18-24 under 35 U.S.C. § 103(a) as being unpatentable over United States Patent No. 5,926,816 to *Bauer et al.* (hereafter "*Bauer*").

The Applicant has concurrently filed a Notice of Appeal of the Examiner's rejection of Claims 2-8, 10-16, and 18-21. The Applicant has filed this Amendment after final rejection pursuant to 37 C.F.R. § 1.116 to correct certain typographical errors in the Specification and to better present the patent application for consideration on appeal.

It is respectfully submitted that the Examiner has misinterpreted the *Bauer* reference.

Please consider that the claims of this Application are directed to a data synchronization apparatus that comprises a “bulk copy controller” and an “update controller” that “operate substantially concurrently.” The Applicants are not claiming a “server” and a “client” that operate substantially concurrently.

On Page 2 of the Office Action the Examiner stated that:

Both client and server [of *Bauer*] have their own synchronizer (See, Column 8 line 3-45). This is referred same as “update controller” that applicant claim (See column 2 line 46-47).

The *Bauer* client database synchronizer 27x and the *Bauer* server database synchronizer 17 perform the function of Applicant’s “update controller 295.”

On Page 3 of the Office Action the Examiner stated that:

Bauer does not clearly disclose the limitation of server and client both operate substantially concurrently. However, referring to Fig. 6A and 6B, column 4 line 40-55, column 11 line 23 - column 13 line 60, wherein *Bauer* teaches that proper synchronization should be frequently verified, and the communication between client and server, it is clear that the claimed provision is inherent. The process that occurs between two checkpoints can be referred as “operate substantial concurrently”. Nonetheless, to expedite prosecution, even if the limitation of the above were not inherent, it would have been obvious to one with ordinary skill in the art the time the invention was made to include such a step in order to generate a system that can operate concurrently.

The function described by *Bauer* using Fig. 6A and Fig. 6B is the function performed by the Applicant’s “update controller 295.” The operation of the *Bauer* device does not disclose, suggest or even hint at the Applicant’s claimed “bulk copy controller 290” and “update controller 295” operating substantially concurrently. The portion of *Bauer* cited by the Examiner simply describes a method for sending data from a server to a client.

The Examiner asserted that the limitation of operating a server and a client “substantially concurrently” is either “inherent” or “obvious.” The teaching of *Bauer* refutes such an assertion. Specifically, *Bauer* teaches that “[h]owever, for any refresh messages to be applied to the client table Tc, the client table Tc and the before-image log table Tb must have identical data. This implies that all current updates for a table T must be propagated to the server before a refresh request can be sent. Furthermore, refresh messages cannot be applied at the client for table T in any modifications had been made to the client table Tc since that last propagate.” (Emphasis added) (*Bauer*, Col. 12, Lines 13-20).

The term “before” in the above-emphasized portion shows that there is no concurrent operation of the server and the client in *Bauer*. That is, the server and the client of *Bauer* do not operate concurrently. First the client determines what client modifications have occurred since the client was last synchronized by the server. Then the client sends the modifications to the server. The server then determines what modifications have occurred at the server since the server last synchronized the client. The server then resolves any data conflicts and sends the modifications to the client as refresh data. (*Bauer*, Col. 12, Line 64 to Col. 13, Line 8). In *Bauer*, there is no concurrent operation between the server and the client. A careful reading of *Bauer* discloses that *Bauer* is directed to periodic (*i.e.*, non-concurrent) updating of server records based upon changed client records.

Bauer fails to teach, suggest or even hint at the Applicant’s bulk copy controller and update controller and the fact that they operate substantially concurrently. *Bauer* further fails to provide any

suggestion or motivation, either in *Bauer* alone or with the knowledge generally available to one of ordinary skill in the art, to modify *Bauer* to arrive at the Applicant's invention.

The only way one can arrive at the present invention is by looking backward from *Bauer* at the Applicant's invention, and, even then, one cannot make the *Bauer* server and clients perform bulk copies of data records from source data files to copy data files while an update controller substantially concurrently operates to detect changes in a data record previously copied and copying the changed data records from the source data file to the copy data file. It cannot be said that one of ordinary skill in the pertinent art would be presumed to know of the teachings of *Bauer* and could solve the same or a similar problem as that the Applicant addresses. *EWP Corp. v. Reliance Universal, Inc.*, 755 F.2d 898, 906-07, 225 U.S.P.Q. 20, 25 (Fed. Cir.), *cert. denied*, 474 U.S. 843 (1985); *In re Sernaker*, 702 F.2d 989, 995, 217 U.S.P.Q. 1, 6 (Fed. Cir. 1983). The requisite motivation does not stem from any of these teachings, from the perspective of one of ordinary skill in the art, to arrive at the Applicant's invention. *Uniroyal Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 5 U.S.P.Q.2d 1434 (Fed. Cir. 1988).

Simply stated, Claim 2 is not *prima facie* obvious. As between Claim 2 and *Bauer*, Independent Claims 10 and 18 contain analogous limitations to those found in independent Claim 2. Claims 10 and 18 are therefore also not *prima facie* obvious. Dependent Claims 3-8, 11-16 and 19-24 include the limitations of their respective base and intervening claims. Therefore, the Applicant respectfully submits that Claims 2-8, 10-16 and 18-24 are patentable over *Bauer*.

SUMMARY

For the reasons given above, the Applicant respectfully requests reconsideration and allowance of pending claims and that this Application be passed to issue. If any outstanding issues remain, or if the Examiner has any further suggestions for expediting allowance of this Application, the Applicant respectfully invites the Examiner to contact the undersigned at the telephone number indicated below or at *wmunck@davismunck.com*.


The Commissioner is hereby authorized to charge any additional fees connected with this communication or credit any overpayment to Davis Munck Deposit Account No. 50-0208.

Respectfully submitted,

DAVIS MUNCK, P.C.

Date: Sept. 9, 2002

P.O. Drawer 800889
Dallas, Texas 75380
Phone: (972) 628-3600
Fax: (972) 628-3616
E-mail: *wmunck@davismunck.com*



William A. Munck
Registration No. 39,308

APPENDIX "A"

VERSION WITH MARKINGS TO SHOW CHANGES MADE

The Specification at Page 5, Line 22 to Page 6, Line 4, has been amended as follows:

According to a still further embodiment of the present invention, the update controller is capable of determining that the copy data file is on line and is capable of activating the bulk copy controller by setting at least one synchronization descriptor in the source data file to a zero value.

The Specification at Page 16, Line 17 to Page 17, Line 6, has been amended as follows:

Next, bulk copy controller 290 copies data records one at a time from first source data table 210 to first copy data table [261] 260. After each source data record is transferred, bulk copy controller 290 increments the synchronization descriptor associated with the source data record in memory 200 (process step 310). After a specific source table is completely copied and transferred, bulk copy controller 290 begins copying and transferring data from the next data table in the source data file. Bulk copy controller 290 continues to increment the synchronization descriptor associated with each data record until each source data table in memory 200 has been copied and transferred to the appropriate data table in memory 250 (process step 315).

The Specification at Page 17, Line 22 to Page 18, Line 13, has been amended as follows:

When changed data is found in a previously transferred source data record in memory 200, update controller 295 transfers a copy of the changed data record from memory 200 to the appropriate data record location in memory 250 (process step 330). By way of example, if SD1 = 4, then update controller 295 detects write operations in Record 1 through Record 4 in data table 210 and copies the changed data to Record 1 through Record 4 in data table [250] 260. When the bulk copy process is first initiated, update controller 295 has no data to examine for change since new source data has not been transferred to copy memory 250, as indicated by the zero value in all synchronization descriptors in memory 200. However, the number of data records examined by update controller 295 increases with each data record copied and transferred by bulk copy controller 290.

The Specification at Page 18, Line 14 to Line 21, has been amended as follows:

Once the bulk copy and transfer of all source data records is complete, bulk copy controller 290 examines memory 200 for all zero values in the synchronization descriptors in memory 200. Simultaneously, update controller 295 continuously examines data records in memory 200 for changes and copies changed data records to memory 200, while also determining if the updating process needs to stop (other side offline) and then detecting when a bulk copy process can be performed (other side back online).